Supporting Information

Heeger, *PNAS* 2017

Extended discussion

The theoretical framework presented in this paper, of course, includes components previously proposed in computational/theoretical neuroscience, image processing, computer vision, statistics, and machine learning with artificial neural networks. Because of space limitations, a number of influential papers were not cited in the main text (1-27).

Variants and extensions

There are a number of variations of the computational framework, depending on the network architecture, output nonlinearity, and optimization algorithm.

The neural responses were modeled as dynamical processes that minimize an energy function over time, via gradient descent. But other optimization algorithms might converge more rapidly while providing a better characterization of empirical measurements of neural dynamics.

The architecture (number of layers, number of channels per layer, interconnectivity between channels in adjacent layers) and spatial weights determine the selectivity of the neurons. I used an architecture and spatial weights that computed motion (Fig. 7 of the main text), but different choices would extract different features (or statistics) of the input. There need not be a strict hierarchy so that, for example, there can be feedforward connections from V1 to V2 to V4 and also a parallel feedforward connection directly from V1 to V4.

The temporal weights determine the predictive basis functions. I used temporal weights that conferred a set of predictive basis functions that are damped oscillators of various temporal frequencies, but different temporal weights might be used instead, corresponding to different predictive basis functions. An obvious variation is to replace the pair of temporal weights (w_m in Eq. **3** of the main text) with a matrix of weights so that the responses of each neuron are predicted over time by a weighted sum of a large number other neurons, including neighboring neurons in the same channel (e.g., that respond to stimuli at nearby spatial locations), and neurons from different channels in the same layer (e.g., that respond preferentially to different stimulus features). This is similar to classic recurrent network models of working memory that maintain a memory representation with a self-sustaining pattern of persistent activity (28-32), and also to some models of perceptual organization, segmentation, and grouping (33-36).

The convolutions can be replaced with an equivalent computation that encompasses the physiological diversity across individual neurons. The convolution weights that determine the selectivity of each neuron in each channel should be thought of as a basis set, with the first basis function equal to the first channel's weights, etc. For some basis sets and for some output nonlinearities (e.g., squaring), any invertible linear transform of the basis set can be substituted (37-39). A different invertible linear transform can be applied at each location, thereby allowing the weights to be different from one location to the next (and explaining the diversity of tuning properties of neurons), without changing the nature of the representation.

In the current implementation, the same neurons perform both inference and prediction, but an alternative implementation of the same principles would be to have two separate subpopulations of neurons. The first subpopulation would be responsible for inference (minimizing both terms in Eq. **3** of the main text), while the second subpopulation would be continuously predicting forward in time, based on the responses of the first subpopulation (minimizing only the second term in Eq. **3** of the main text). These two subpopulations of neurons might be in the same cortical circuit or the prediction subpopulation of neurons might be in a different brain area.

Normalization and other output nonlinearities

The examples in this paper, only for the sake of simplicity, used quadratic output nonlinearities, but a computation called "the normalization model" has been found to be a better model (both theoretically and empirically) of the output nonlinearity (40). I developed the normalization model 25 years ago to explain stimulus-evoked responses of individual neurons in V1 (41, 42). The model has since been applied to explain physiological measurements of neural activity in a wide variety of neural systems (43-60), and behavioral/ perceptual analogs of those physiological phenomena (e.g., 53, 54, 59, 61-63). The defining characteristic of normalization is that the feedforward drive underlying the response of each neuron is raised to a power (e.g., squaring) and divided by a factor that includes a sum of activity of a pool of neurons, analogous to normalizing the length of a vector (see below, Eq. **S3**). Squaring can be computed with a pair of neurons that have complementary weights (flipped in sign), each of which is half-squared (halfwave rectified and squared) and then summed (41). The half-squaring can be approximated by rectification with a high threshold (64, 65).

It has been known since the normalization model was first introduced that normalization can be implemented in a recurrent neural circuit with biophysically-plausible mechanisms (40, 42, 55, 66-68), but only recently has there been progress in elucidating the cellular and biophysical mechanisms underlying normalization. Normalization is implemented by GABA-mediated presynaptic inhibition in the olfactory system of the fruit fly (48, 69). Normalization in mammalian cortex, however, does not rely on GABA inhibition (70), but rather is caused by a decrease in excitation (71). That is, the mechanisms underlying normalization are different in different neural systems.

Sigmoids, rectified linear units, and max pooling are alternative output nonlinearities, common/popular in computational neuroscience and machine learning, that are each related to normalization. The normalization model, because of the division, confers a saturating (sigmoidal) response as a function of the amplitude of the inputs. A rectified linear unit computes a linear sum of its inputs and subtracts a constant bias, followed by halfwave rectification. The bias acts like a high threshold, that approximates a power function with different values of the bias corresponding to different powers (64, 65). Max pooling (also called softmax) transmits the most active response among a set of inputs (72). Max pooling can be approximated by normalization (73).

Learning the prior

The priors can be learned. For a prior that constitutes a permanent feature of the environment, an elegant solution is to adjust the convolution weights (i.e., "warp" the tuning curves) to match the statistics of the environment (74). The current theory handles the priors in a complementary way. Some priors, rather than being a permanent feature of the environment, are instead context-specific (e.g., matched to a particular task). The cue combination network (Fig. 5 of the main text) provides an example. What I have in mind is that this cue combination network is embedded in a larger hierarchical network. The target values for the responses $\hat{y}$ are learned as the mean responses of the neurons, averaged across a series of practice/training trials in which the cues are consistent with one another (no cue conflict), and both cues are reliable (i.e., with large stimulus strengths). These learned target responses propagate up the hierarchy, transformed to an abstract representation, and stored in memory. Just before each trial of the task, this abstract representation is recalled from memory at the top of the hierarchy, and the state of the network is set to behave like a generative model so the remembered prior is propagated via the feedback drive to a sensory representation, i.e., to reconstruct the target response values. The state is then switched so that this sensory representation of the priors is combined with incoming sensory information to perform inference.

Brain states, neuromodulators, and oscillatory activity

The values of the state parameters (α and λ) determine whether neural responses are driven bottom-up, top-down, or a combination of the two. These parameters also control whether the neurons are primarily processing sensory inputs that occurred in the past versus predicting the future. There is evidence that acetylcholine (ACh) plays a particular role in modulating the trade-off between bottom-up sensory input versus top-down signals related to expectancy and uncertainty (e.g., 75). It has also been hypothesized that ACh signals when bottom-up sensory inputs are known to be reliable (76, 77). Consequently, it is reasonable to hypothesize that α and/or λ might be controlled (at least in part) by ACh. Although ACh is released broadly throughout the cortex, its effect can be regionally specific (78), possibly offering a mechanism for how the values of the state parameters can differ across the hierarchy of brain areas.

In addition, there is considerable evidence that attention modulates the gain of neural responses (51), suggesting that α might be controlled also by attention, perhaps through the feedback drive (see paragraph above about learning the prior) or through a different set of feedback connections that modulate the gain of the convolutions v.

Neuromodulators might also control changes in state to enable learning. During inference, the neural responses are computed dynamically with fixed weights. During learning, the weights are adjusted to minimize the difference between the predicted and the actual neural responses. Neuromodulators might indicate when it is appropriate to adjust the weights (e.g., moments in time corresponding to prediction errors). Dopamine, for example, has been identified as signaling reward prediction-error (79).

According to the theory, exploration depends on neural response variability, which might be controlled (at least in part) by noradrenaline (NA). Specifically, I added non-stationary noise to the simulated neural responses to implement a kind of stochastic optimization. I speculate that the time course of spontaneous NA fluctuations might contribute to the time-varying standard deviation of this non-stationary noise process. Subthreshold fluctuations in NA over time (as assessed by measuring pupil dilation) affect neural response variability (80). Neural response variability exhibits an inverted U-shaped curve as a function of membrane potential depolarization such that responses are most reliable for an intermediate level of depolarization and less reliable when the neural membrane potential is either too close or too far from spike threshold. Neural membrane-potential depolarization and pupil size both depend on NA. For example, NA fluctuations might exhibit a $1/f$ amplitude spectrum (81). Such a noise process can be computed by integrating white noise over time (analogous to the position of a particle undergoing Brownian motion); doing so with a leaky integrator is biologically plausible given the ubiquity of neural integrators (31, 32). It has been hypothesized that NA signals when something unexpected has occurred (76, 77), which would, according to the present theory, transiently increase the noise variance to explore alternative interpretations. NA has also been linked to alternations (i.e., exploration) during bistable perception (82), an observation that might be explained by the current theory if perception is stable when the neural response variability is low and prone to alternations when response variability is high.

This non-stationary noise process might also contribute to variability over time in behavioral performance. Measurements of behavioral performance as a function of arousal exhibit an inverted U-shaped function, which is hypothesized to be caused by the relationship between NA and neural response variability (80, 83-85). It has been reported, for example, that residual reaction time (after subtracting the mean reaction time for any given experimental condition) exhibits a $1/f$ power spectrum for a variety of tasks (86). Behavioral measures of timing and tapping also exhibit $1/f$ power spectra (87, 88).

Neural response variability might also be controlled (in part) by oscillations in brain activity, pseudo-periodic fluctuations in neural membrane potential, correlated across large populations of neurons. Such brain oscillations are readily observed with EEG, a well-known example of which is so-called alpha activity (~10 Hz). Subthreshold fluctuations in neural membrane potential affect neural response variability, as summarized above (80). I presume that such fluctuations have an impact on the reliability of stimulus-evoked activity with little or no impact on the mean responses (i.e., that the fluctuations are small in any given neuron but that they are evident in EEG recordings which measures the correlated component of the membrane potential fluctuations across a large population of neurons). So I hypothesize that oscillations in brain activity might contribute to stochastic optimization for exploring alternative perceptual and/or cognitive interpretations. The oscillation phase corresponding to minimal response variability would correspond to the more stable percepts and the phase corresponding to maximum response variability would correspond to less stable percepts. These periodic fluctuations in response variability (in service of optimization) might, therefore, explain the empirical evidence for perceptual rhythms, i.e., that perception and perceptual performance fluctuate periodically and depend on the frequency and phase of oscillatory activity (89).

Methods and derivations

Feedforward convolutional neural net

Deep convolutional neural nets have an architecture that is based on a common model of sensory processing in the visual system, comprising a feedforward (pipeline processing) hierarchy of stages each comprising a bank of linear filters following by an output nonlinearity (Figs. 2A,B of the main text). This hierarchy of computations can be expressed as follows:

$$y_{jn}^{(i)} = \rho_z\left(v_{jn}^{(i)}\right)$$
$$v_{jn}^{(i)} = \sum_{q=1}^{N^{(i-1)}} \sum_k w_{jknq}^{(i-1)} y_{kq}^{(i-1)} \quad . \qquad \textbf{[S1]}$$

The values of y are the responses (proportional to firing rates) of the neurons in each layer, v are the outputs of the linear weighted sums, w are weight matrices, and ρ_z is the output nonlinearity. The superscript (i) specifies the layer in the hierarchy; $y^{(0)}$ are the inputs to the multi-layered hierarchy. The subscripts n and q specify each of the channels in a layer, where $N^{(i)}$ is the number of channels in layer (i). The subscripts j and k specify the different neurons in a channel. The values of w_{jknq} specify a matrix of weights connecting the k^{th} neuron in channel q of layer $(i\text{-}1)$ to the j^{th} neuron of channel n of layer (i). For all neurons in a channel, the weight matrices are assumed to be spatially shifted copies of one another (i.e., performing a spatial convolution, optionally with spatial subsampling). I have included the subscripts n and q in w_{jkqn} only to clarify that the weights are different for different channels.

The examples in this paper use either linear outputs or quadratic output nonlinearities:

$$\rho_z(v) = v$$
$$\rho_z(v) = \frac{1}{2}v^2 \quad . \qquad \textbf{[S2]}$$

Normalization is a more sophisticated model of the nonlinearity (40). The defining characteristic of normalization is that the response of each neuron is divided by a factor that includes a sum of activity of a pool of neurons:

$$\rho_Z\left(v_{jn}^{(i)}\right) = \frac{\left(v_{jn}^{(i)}\right)^2}{\sum_q \sum_k \beta_{kqn}^{(i)} \left(v_{kq}^{(i)}\right)^2 + \left(\sigma^{(i)}\right)^2} \quad . \qquad \textbf{[S3]}$$

The summation in the denominator is a weighted sum (i.e., local average) over neurons in the same layer with weights β. For each neuron j in channel n, these weights β_{kq} are assumed to be spatially shifted copies of one another (i.e., performing a spatial convolution). I have included the subscript n in β_{kqn} only to clarify that the weights β_{kq} are different for different channels. The constant σ determines the contrast gain (the contrast of the visual stimulus that evokes half the maximal response).

Theory of Cortical Function

I hypothesize that neural responses minimize an energy function (or optimization criterion) across all neurons in all channels and layers (and a summation over time can also be included, see below):

$$E = \sum_{i=1}^{L} \sum_n \sum_j \alpha^{(i)} \lambda^{(i)} \rho_l\left(y_{jn}^{(i)} - z_{jn}^{(i)}\right)$$
$$+ \sum_{i=1}^{L} \sum_n \sum_j \alpha^{(i)} \left(1 - \lambda^{(i)}\right) \rho_p\left(y_{jn}^{(i)} - \hat{y}_{jn}^{(i)}\right) \qquad \textbf{[S4]}$$
$$z_{jn}^{(i)} = \rho_z\left(v_{jn}^{(i)}\right)$$
$$v_{jn}^{(i)} = \sum_{q=1}^{N^{(i-1)}} \sum_k w_{jknq}^{(i-1)} y_{kq}^{(i-1)} \quad .$$

This is a generalization of Eq. **1** of the main text with multiple channels in each layer and a flexible choice for the output nonlinearities and cost functions. The values of y are again the neural responses (proportional to firing rates). The values of v are again the outputs of the linear weighted sums from the previous layer. The values of z are now the outputs after the nonlinearity (unlike the more common formulation above in which y are the outputs after the nonlinearity). The function ρ_z is again the output nonlinearity (Eq. **S2**). The values of $\hat{y}$ in the second term represent a prior (or expectation) for the responses. These variables (y, x, v, z, and $\hat{y}$) are each functions of time because the inputs change over time with the sensory input. The functions ρ_l and ρ_p are cost functions, which are quadratic for the examples in this paper:

$$\rho_l(u) = \frac{1}{2}u^2 \qquad \rho_p(u) = \frac{1}{2}u^2 \quad , \qquad \textbf{[S5]}$$

although other cost functions could be readily substituted. The values of α and λ ($0 < \lambda < 1$) are state parameters that determine the tradeoffs between the two terms in the energy function at each layer.

The neural responses are modeled as dynamical processes that minimize this energy function over time (dropping the channel subscript n to simplify notation):

$$\tau \frac{dy_j^{(i)}}{dt} = -\frac{dE}{dy_j^{(i)}} \quad \textbf{[S6]}$$

The derivative of the energy function with respect to each neuron's response (using quadratic output nonlinearities and quadratic cost functions) is:

$$\frac{dE}{dy_j^{(i)}} = \alpha^{(i)}\lambda^{(i)}\left(y_j^{(i)} - z_j^{(i)}\right) + \alpha^{(i)}\left(1-\lambda^{(i)}\right)\left(y_j^{(i)} - \tilde{y}_j^{(i)}\right) + \sum_k \frac{dE}{dz_k^{(i+1)}}\frac{dz_k^{(i+1)}}{dy_j^{(i)}} \quad \textbf{[S7]}$$

Combining the previous two equations yields the following dynamical system in which each neuron's response is updated over time:

$$\begin{aligned} \tau \frac{dy_j^{(i)}}{dt} &= -\alpha^{(i)}\lambda^{(i)} f_j^{(i)} + \alpha^{(i+1)}\lambda^{(i+1)} b_j^{(i)} - \alpha^{(i)}\left(1-\lambda^{(i)}\right) p_j^{(i)} \\ f_j^{(i)} &= y_j^{(i)} - z_j^{(i)} \\ b_j^{(i)} &= \sum_k \left[y_k^{(i+1)} - z_k^{(i+1)}\right] v_k^{(i+1)} w_{kj}^{(i)} \\ p_j^{(i)} &= y_j^{(i)} - \hat{y}_j^{(i)} \end{aligned} \quad \textbf{[S8]}$$

This is the same as Eq. **2** of the main text except that I have included factors of 1/2 in the quadratic output nonlinearity and the quadratic cost function. As noted in the main body of the paper, the first term in this expression is the feedforward drive f; with only this term the neural responses would be the same as the feedforward model outlined above (i.e., $y = z$). The second term is the feedback drive b; this term drives the responses according to the mismatch between the responses at the next layer, i+1, and the feedforward drive from the i^{th} layer. The third term is the prior drive p; with only this term the neural responses would be driven to the value of the prior (i.e., $y = \hat{y}$). The value of τ is a time constant.

Feedback connections

As noted in the main body of the paper, the feedback signals are selective for features that are represented at the earlier layer due to the transpose of the weight matrix. A simplified two-layer example illustrates:

$$\begin{aligned} E &= \frac{1}{2}\sum_j \left(y_j^{(2)} - z_j^{(2)}\right)^2 \\ z_j^{(2)} &= \frac{1}{2}\left(\sum_k w_{jk} y_k^{(1)}\right)^2 = \frac{1}{2}\left(v_j^{(2)}\right)^2 \\ v_j^{(2)} &= \sum_k w_{jk} y_k^{(1)} \\ \frac{dE}{dy_k^{(1)}} &= \sum_j \frac{dE}{dz_j^{(2)}}\frac{dz_j^{(2)}}{dy_k^{(1)}} = -\sum_j \left(y_j^{(2)} - z_j^{(2)}\right) w_{jk} v_j^{(2)} \end{aligned} \quad \textbf{[S9]}$$

In the form of a matrix tableau:

$$\textbf{[S10]}$$

$$\begin{pmatrix} \vdots \\ v_j^{(2)} \\ \vdots \end{pmatrix} = \begin{pmatrix} \ddots & & \\ & w_{jk} & \\ & & \ddots \end{pmatrix} \begin{pmatrix} \vdots \\ y_k^{(1)} \\ \vdots \end{pmatrix}$$

$$\begin{pmatrix} \vdots \\ \dfrac{dE}{dy_k^{(1)}} \\ \vdots \end{pmatrix} = -\begin{pmatrix} \ddots & & \\ & w_{kj} & \\ & & \ddots \end{pmatrix} \begin{pmatrix} \vdots \\ \left(y_j^{(2)} - z_j^{(2)}\right) v_j^{(2)} \\ \vdots \end{pmatrix}.$$

The feedforward drive depends on $v_j^{(2)}$, which is computed as a weighted sum of the layer 1 responses $y_k^{(1)}$ with weights w_{jk}. The gradients of the energy function $dE/dy_k^{(1)}$, which determine the feedback drives, are computed as a weighted sum of the mismatch between the responses and the feedforward drive $(y_j^{(2)}-z_j^{(2)})\ v_j^{(2)}$ using the transpose of the weight matrix w_{kj}.

Inference (Fig. 3)

For each of the simulation results in Fig. 3 of the main text, the input, the prior $\hat{y}$, and the network state (determined by the values of λ and α) were all held constant over time. The responses of the neurons were initialized to small, random values ($0 < y < 0.1$) at time $t = 0$. The responses were computed with Eq. **2** of the main text (time constant: $\tau = 5$ ms; time step: $\Delta t = 1$ ms), and the values were clipped ($0 < y < 1$) after each iteration.

Exploration (Fig. 4)

The responses were again computed with Eq. **2** of the main text (time constant: $\tau = 5$ ms; time step: $\Delta t = 10$ ms), the values were again clipped ($0 < y < 1$), and noise was added to each neuron's response at each time step. The noise was statistically independent across neurons and over time, but non-stationary. All neurons had the same noise standard deviation at each moment in time, but the noise standard deviation varied over time. Specifically, the time course of the standard deviation had a $1/f$ amplitude spectrum for frequencies greater than ~1 Hz. The noise process was computed by taking Gaussian white noise and filtering it with a leaky integrator (i.e., a first-order differential equation or exponential low pass filter) with time constant = 100 ms. The noise added to each neuron at each time point was drawn (independently for each neuron and each time point) from a normal distribution with the corresponding standard deviation.

One-layer time-series prediction (Fig. 6)

The one-layer time-series prediction network (Fig. 6 of the main text) optimized the following energy function:

$$E = \frac{1}{2}\sum_t \lambda(t)\left[\left(\sum_m \mathrm{Re}\left(y_m^{(1)}(t)\right)\right) - y^{(0)}(t)\right]^2 + \frac{1}{2}\sum_t \left(1-\lambda(t)\right)\left[\sum_m \left|y_m^{(1)}(t) - \hat{y}_m^{(1)}(t)\right|^2\right] \quad \textbf{[S11]}$$

$$\hat{y}_m^{(1)}(t) = y_m^{(1)}(t-\Delta t)\, w_m^{(1)}$$

$$w_m^{(1)}(\Delta t) = e^{i2\pi\omega_m^{(1)}\Delta t}.$$

This is a different way of writing Eq. **3** of the main text using the notational convenience of complex numbers and complex exponentials (instead of sines and cosines). It is a global optimization criterion; the summation is over all neurons and over time. The values of y_m are the complex-valued responses of a subpopulation of neurons that share the same input $y^{(0)}$, the values of ω_m specify the frequencies of the predictive basis functions, w_m are temporal weights (a pair of numbers for a each ω_m), and Δt is a discrete time step. The complex values can be represented by the responses of a pair of neurons (Eq. **3** of the main text), but the complex-exponential notation is convenient. The state parameter λ can change over time.

The derivative of E with respect to $y_m(t)$, can be used to find a local minimum of E by gradient descent:

$$\Delta y_m^{(1)}(t) = -r\frac{\partial E}{\partial y_m^{(1)}(t)} = -r\left[f_m^{(1)}(t) + p_m^{(1)}(t)\right], \quad \textbf{[S12]}$$

where f_m is the feedforward drive (note that there is no feedback drive in this one-layer example), p_m is the prior drive, r specifies a step size, and $y_m(t)$ is updated simultaneously for all time points t. I used Eq. **S12** to implement a batch algorithm, to compute the global minimum for all neurons and all time samples (Fig. 6 of the main text). This batch algorithm updated all of the neural responses at all time samples repeatedly until it converged. Other optimization algorithms could be used instead; For example, I have implemented an incremental approximation (see below).

The expressions for f_m and p_m depend on whether there is an input (for $t \le 0$) or not (for $t > 0$), and whether or not t is an endpoint (e.g., for a finite duration simulation and/or with an incremental algorithm for which $t = 0$ is always an endpoint because the input for the next time step is in the future). The feedforward drive is:

$$f_m^{(1)}(t) = \lambda\left[\mathrm{Re}\left(\sum_m y_m^{(1)}(t)\right) - y^{(0)}(t)\right], \quad \textbf{[S13]}$$

when there is an input and 0 otherwise. The prior drive is:

$$p_m^{(1)}(t) = \left(1-\lambda(t)\right)\left(2y_m^{(1)}(t) - y_m^{(1)}(t-\Delta t)w_m^{(1)}(\Delta t) - y_m^{(1)}(t+\Delta t)w_m^{(1)}(-\Delta t)\right)$$

$$p_m^{(1)}(t) = \left(1-\lambda(t)\right)\left(y_m^{(1)}(t) - y_m^{(1)}(t+\Delta t)w_m^{(1)}(-\Delta t)\right) \quad \textbf{[S14]}$$

$$p_m^{(1)}(t) = \left(1-\lambda(t)\right)\left(y_m^{(1)}(t) - y_m^{(1)}(t-\Delta t)w_m^{(1)}(\Delta t)\right),$$

when t is not an endpoint, when t is the first time sample, and when t is the last time sample, respectively.

For the simulation results (Fig. 6 of the main text), the input was a sum of two sinusoids (2 Hz, amplitude 1; 8 Hz, amplitude 1/2) for past time ($t \le 0$) and nonexistent for future time ($t > 0$). I.e., the first term of E in Eq. **S11** was set to 0 (ignoring the input entirely) for $t > 0$. This could be implemented with two separate subpopulations of neurons, one of which minimizes both terms in Eq. **S11** and is responsive to the input, while the second subpopulation minimizes only the second term in Eq. **S11** and is continuously predicting forward in time (see *Variants and extensions*). Regardless, this is different from setting the input to 0 and minimizing both terms of E. If the input was set to 0 for $t > 0$ (rather than ignoring it entirely) then the responses decayed over time; the value of λ determined rate at which the responses decayed (see below).

Unlike the examples in Figs. 3 and 4 of the main text, the responses were not clipped. The negative values for the responses can be accommodated with positive firing rates by replacing each quadrature pair with 4 neurons, each with halfwave-rectified responses and 4 different temporal phases offset by 90°.

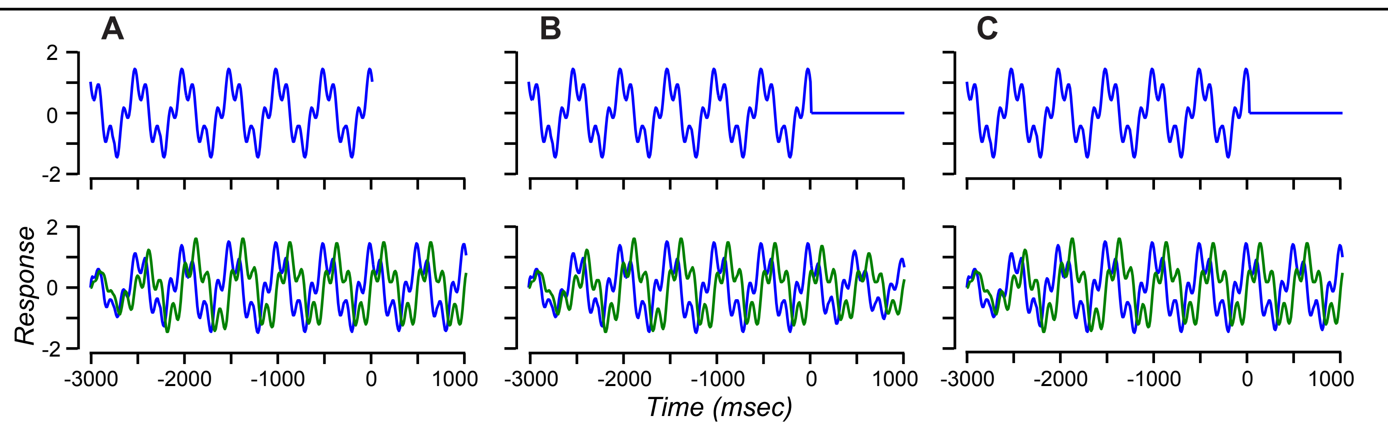

Fig. S1. One-layer time-series prediction with incremental algorithm. Compare with Fig. 6A of the main text. **A.** Input is a sum of two sinusoids for past time ($t \le 0$) and nonexistent for future time ($t > 0$). **B,C.** Input is 0 for $t > 0$. Top row, input. Bottom row, output. Blue curves in the bottom row, sum of the responses of the neurons representing the real parts of y_m. Green curves, sum of neural responses representing the imaginary parts of y_m. **A,B.** State: $\boldsymbol{\lambda} = 0.1$ for $t \le 0$ and $\boldsymbol{\lambda} = 0.01$ for $t > 0$ (same as Fig. 6 of the main text). **C.** State: $\boldsymbol{\lambda} = 0.1$ for $t \le 0$ and $\boldsymbol{\lambda} = 0.001$ for $t > 0$.

An incremental (causal) algorithm was also implemented (Fig. S1), analogous that in Eq. **S8**. The prior drive for the incremental algorithm used the 3rd line of Eq. **S14**, so that the change in responses depended on only the present input and the present and past responses. In practice, fewer than 10 iterations were needed for each time step of the incremental algorithm. The results depended on whether the input was nonexistent for $t > 0$ (Fig. S1A) or 0 for $t > 0$ (Figs. S1B,C). For nonexistent input, the first term of E in Eq. **S11** was set to 0 (ignoring the input entirely) for $t > 0$. If the input was set to 0 for $t > 0$ (rather than ignoring it entirely) then both terms of E were minimized, and the responses decayed over time; the value of λ determined rate at which the responses decayed (Figs. S1B,C).

Multi-layer prediction of periodic motion (Fig. 7)

The multi-layer prediction network optimized the following energy function:

$$\begin{aligned}
E &= \frac{1}{2}\sum_{i=1}^{L}\sum_{n}\sum_{t}\alpha^{(i)}(t)\lambda^{(i)}(t)\left[\left(\sum_{m}\mathrm{Re}\left(y_{nm}^{(i)}(x,t)\right)\right) - z_{n}^{(i)}(x,t)\right]^2 \\
&\quad + \frac{1}{2}\sum_{i=1}^{L}\sum_{n}\sum_{t}\alpha^{(i)}(t)\left(1-\lambda^{(i)}(t)\right)\left[\sum_{m}\left(y_{nm}^{(i)}(x,t) - \hat{y}_{nm}^{(i)}(x,t)\right)^2\right] \\
\hat{y}_{nm}^{(i)}(x,t) &= y_{nm}^{(i)}(x,t-\Delta t)\; w_{m}^{(i)} \\
w_{m}^{(i)}(\Delta t) &= e^{i 2\pi\omega_{m}^{(i)}\Delta t}
\end{aligned} \qquad \textbf{[S15]}$$

Here, I dropped the subscript j and instead use x to denote the different neurons in each channel in terms of the spatial locations of their receptive fields. The rest of the notation is defined above (Eqs. **S4**, **S11**). It is again a global optimization criterion; the summation is over all neurons in all channels and all layers, and over time.

To denote the specific multi-layer motion-prediction network (Fig. 7 of the main text), it is helpful to break it down and write each layer separately. The total energy was the sum of the energies in each layer:

$$E = E^{(1)} + E^{(2)} + E^{(3)}. \qquad \textbf{[S16]}$$

Layer 1 had one channel:

$$\begin{aligned}
E^{(1)} &= \frac{1}{2}\sum_{t}\alpha^{(1)}\lambda^{(1)}\left[\mathrm{Re}\left(\sum_{m}y_{m}^{(1)}\right) - y^{(0)}\right]^2 \\
&\quad + \frac{1}{2}\sum_{t}\sum_{m}\alpha^{(1)}\left(1-\lambda^{(1)}\right)\left(y_{m}^{(1)} - \hat{y}_{m}^{(1)}\right)^2
\end{aligned} \qquad \textbf{[S17]}$$

The values of $y^{(0)}(x,t)$ are the output of a simplified model of retinal processing consisting of a temporal filter at each spatial location (see below for details). The responses y are functions of both space and time, and the state parameters α and λ also change over time, but I have dropped x and t from this equation (and in most of those that follow) to simplify the notation.

Layer 2 had four channels:

$$\begin{aligned}
E^{(2)} &= \frac{1}{2}\sum_{t}\sum_{n}\alpha^{(2)}\lambda^{(2)}\left[\mathrm{Re}\left(\sum_{m}y_{nm}^{(2)}\right) - z_{j}^{(2)}\right]^2 \\
&\quad + \frac{1}{2}\sum_{t}\sum_{n}\sum_{m}\alpha^{(2)}\left(1-\lambda^{(2)}\right)\left(y_{nm}^{(2)} - \hat{y}_{nm}^{(2)}\right)^2 \\
z_{n}^{(2)}(x,t) &= \frac{1}{2}\left(v_{n}^{(2)}(x,t)\right)^2 \\
v_{n}^{(2)}(x,t) &= \sum_{\xi}\left[w_{n1}^{(1)}(\xi - x)\sum_{m}\mathrm{Re}\left(y_{m}^{(1)}(x,t)\right)\right] \\
&\quad + \sum_{\xi}\left[w_{n2}^{(1)}(\xi - x)\sum_{m}\mathrm{Im}\left(y_{m}^{(1)}(x,t)\right)\right],
\end{aligned} \qquad \textbf{[S18]}$$

The last line expresses v_n as a sum of convolutions, where n indexes the 4 channels, and w_{n1} and w_{n2} are the spatial weights of the convolution kernels (Fig. 7C of the main text; Eq. **S23**). The derivatives, used for gradient descent, are:

$$\begin{aligned}
\frac{dE^{(2)}}{dz_{n}^{(2)}} &= -\alpha^{(2)}\lambda^{(2)}\sum_{n}\left[\left(\sum_{m}y_{nm}^{(2)}\right) - z_{n}^{(2)}\right] \\
\frac{dz_{n}^{(2)}(x,t)}{dy_{m}^{(1)}(x,t)} &= v_{n}^{(2)}(x,t)\frac{dv_{n}^{(2)}(x,t)}{dy_{m}^{(1)}(x,t)} \\
&= v_{n}^{(2)}(x,t)\left[w_{n1}^{(1)}(x) + i w_{n2}^{(1)}(x)\right].
\end{aligned} \qquad \textbf{[S19]}$$

Layer 3 had two channels:

$$\begin{aligned}
E^{(3)} &= \frac{1}{2}\sum_{t}\sum_{n}\alpha^{(3)}\lambda^{(3)}\left[\mathrm{Re}\left(\sum_{m}y_{nm}^{(3)}\right) - z_{n}^{(3)}\right]^2 \\
&\quad + \frac{1}{2}\sum_{t}\sum_{n}\sum_{m}\alpha^{(3)}\left(1-\lambda^{(3)}\right)\left(y_{nm}^{(3)} - \hat{y}_{nm}^{(3)}\right)^2 \\
z_{1}^{(3)} &= \sum_{m}\mathrm{Re}\left(y_{1m}^{(2)}\right) + \sum_{m}\mathrm{Re}\left(y_{2m}^{(2)}\right) \\
z_{2}^{(3)} &= \sum_{m}\mathrm{Re}\left(y_{3m}^{(2)}\right) + \sum_{m}\mathrm{Re}\left(y_{4m}^{(2)}\right) \\
\frac{dE^{(3)}}{dz_{n}^{(3)}} &= -\alpha^{(3)}\lambda^{(3)}\sum_{n}\left[\left(\sum_{m}y_{nm}^{(3)}\right) - z_{n}^{(3)}\right] \\
\frac{dz_{1}^{(3)}}{dy_{1m}^{(2)}} &= 1 \quad \frac{dz_{1}^{(3)}}{dy_{2m}^{(2)}} = 1 \quad \frac{dz_{2}^{(3)}}{dy_{3m}^{(2)}} = 1 \quad \frac{dz_{2}^{(3)}}{dy_{4m}^{(2)}} = 1,
\end{aligned} \qquad \textbf{[S20]}$$

and the other derivatives of $z_j^{(3)}$ with respect to $y_{km}^{(2)}$ are zero.

The simulation results in Fig. 7 of the main text were computed as follows. The input was a sum of two contrast-modulated sinusoids for past time ($t \le 0$):

$$
\begin{aligned}
s(x,t) &= c(t)\sin\left(2\pi\omega_x x - 2\pi\omega_t t\right) \\
&\quad + \left[1 - c(t)\right]\sin\left(2\pi\omega_x x + 2\pi\omega_t t\right) \\
c(t) &= \frac{1}{2}\left[1 + \cos\left(2\pi\omega_m t\right)\right]
\end{aligned}, \qquad \textbf{[S21]}
$$

where $\omega_m = 1$ Hz was the modulation frequency, $\omega_x = 8$ cycle/deg was the spatial frequency, and $\omega_t = 8$ Hz was the grating temporal frequency, so that the speed of motion was 1 deg/sec. The stimulus was sampled spatially with 120 samples per degree of visual angle (approximately equal to the sampling of cone photoreceptors in the fovea of the primate retina) and with 1 ms temporal sampling.

The input was nonexistent for future time ($t > 0$). As for the simulation in Fig. S1, the responses decayed to 0 over time if the input was set to 0 for $t > 0$ (rather than ignoring it entirely), and the value of λ determined rate at which the responses decayed.

A simplified model of retinal processing was computed as a cascade of exponential low-pass filters (Fig. 7A of the main text):

$$
\begin{aligned}
y^{(0)}(x,t) &= f_3(x,t) - f_5(x,t) \\
\tau_f \frac{df_1(x,t)}{dt} &= -f_1(x,t) + s(x,t) \\
\tau_f \frac{df_{n+1}(x,t)}{dt} &= -f_{n+1}(x,t) + f_n(x,t)
\end{aligned}, \qquad \textbf{[S22]}
$$

where $y^{(0)}$ was the retinal output (i.e., the input to the multi-layer motion-prediction network) at each spatial sample x, and $\tau_f = 12$ ms was the time constant of each of the low-pass filters.

Layer 1. The layer 1 weights were the identity matrix and the output was linear, so that the layer 1 responses were driven to copy the retinal input. Layer 1 comprised a pair of neurons corresponding to each spatial location, all with the same temporal frequency tuning that matched that of the sinusoidal grating ($\omega_m = 8$ Hz). One neuron in each pair represented the real part of the complex-valued responses and the other neuron in each pair represented the imaginary part. For each such pair of neurons, the time-courses of the responses were offset by a 90° phase shift.

Layer 2. The layer 2 weights were constructed from even- and odd-phase spatial Gabor functions (Fig. 7B of the main text). Each of these 2 spatial weighting functions was convolved with the responses of each of the two spatial arrays of layer 1 responses to yield 4 space-time separable combinations. Direction-selective responses were computed as sums and differences of these space-time separable responses (90), resulting in 4 direction-selective channels, two of which were a quadrature pair that preferred leftward motion, and two of which were a quadrature pair that preferred rightward motion. The layer 2 output nonlinearity was squaring. Each of the 4 direction-selective channels was combined with 2 predictive basis functions: 0 Hz and 16 Hz (i.e., twice the temporal frequency in layer 1 because the output nonlinearity was quadratic).

The Gabor functions used for the spatial weights in layer 2 (Fig. 7B of the main text) were:

$$
\begin{aligned}
w_s(x) &= \exp\left(x^2/\sigma^2\right)\sin\left(2\pi\omega_x x\right) \\
w_c(x) &= \exp\left(x^2/\sigma^2\right)\cos\left(2\pi\omega_x x\right)
\end{aligned} \qquad \textbf{[S23]}
$$

$$
\begin{aligned}
w_{11}^{(1)}(x) &= w_s(x) & w_{12}^{(1)}(x) &= w_c(x) \\
w_{21}^{(1)}(x) &= w_c(x) & w_{22}^{(1)}(x) &= -w_s(x) \\
w_{31}^{(1)}(x) &= -w_s(x) & w_{32}^{(1)}(x) &= w_c(x) \\
w_{41}^{(1)}(x) &= w_c(x) & w_{42}^{(1)}(x) &= w_s(x)
\end{aligned},
$$

where $\omega_x = 8$ cycle/deg was the preferred spatial frequency and $\sigma = 1/16$ degrees of visual angle determined the extent of the spatial weights.

Layer 3. There were two channels in layer 3. The feedforward drive for the first channel summed the quadrature pair of leftward-selective layer 2 responses, and summed across space. Likewise, the feedforward drive for the second channel summed the quadrature pair of rightward-selective layer 2 responses, and summed across space. The layer 3 output was again linear. Layer 3 had two predictive basis functions: 0 Hz and 1 Hz (i.e., matching the frequency of periodic motion in the stimulus).

The feedforward processing in this network, with no feedback and no prior (i.e., with λ=1), computed leftward and rightward “motion energy” (41, 90). It is called “motion energy” because it depends on the local (in space, time, orientation, spatial frequency, and temporal frequency) spectral energy of the stimulus. But the term “motion energy” has nothing to do with the energy function that is being minimized (Eq. **S15**). Layer 1 comprised a pair of neurons at each spatial location, with the same temporal frequency tuning. One of the neurons in each pair responded with a copy of the input (provided by the simplified temporal-filtering model of retinal processing). The other neuron responded with a phase-shifted copy of the input. The phase shift emerged because of the quadrature-phase (sine and cosine) temporal weights. As an aside, this solves a problem for models of visual motion perception, which rely on having pairs of neurons that respond with temporal phases offset by 90° (90). The feedforward drive in layer 2 depended on odd- and even-phase spatial weights, and a quadrature pair of temporal filters (the real and imaginary parts of the layer 1 responses), combined according to Eq. **S18**. This yielded four direction-selective channels: a quadrature pair selective for rightward motion and a quadrature pair selective for leftward motion (90). The feedforward drive in layer 3 computed motion energy, a sum of the squared responses of each quadrature pair.

The neural responses corresponding to the global minimum of E were computed for all neurons and all time steps (time step: $\Delta t = 10$ ms), using the “batch” algorithm (see above). There was a second local minimum for which the network predicted that the periodic motion would dissipate, with a clear local maximum separating the two local minima.

Bayesian cue combination (Fig. 5)

The energy function for the cue-combination network was:

$$E\left(\mathbf{y}^{(1)}\right)=\frac{1}{2}\alpha\lambda\sum_n\left(y_n^{(1)}-z_n^{(1)}\right)^2+\frac{1}{2}\alpha(1-\lambda)\sum_n\left(\frac{y_n^{(1)}}{\hat{g}}-\hat{y}_n^{(1)}\right)^2$$
$$\hat{g}=\sum_n y_n^{(1)}$$
$$z_n^{(1)}=w_1 y_{n1}^{(0)}+w_2 y_{n2}^{(0)}$$
$$w_1=\sqrt{\frac{\sigma_2^2}{2\left(\sigma_1^2+\sigma_2^2\right)}}$$
$$w_2=\sqrt{\frac{\sigma_1^2}{2\left(\sigma_1^2+\sigma_2^2\right)}},\qquad\textbf{[S24]}$$

where to $y_{n1}^{(0)}$ and to $y_{n2}^{(0)}$ are the responses of two sets of input neurons and $y_n^{(1)}$ are the responses of the output neurons (Fig. 5A of the main text). Each of the input neurons was tuned for depth, responding most strongly to a preferred depth value (Fig. 5B of the main text). Consequently, each input neuron was from a different channel, indexed by n. (A channel by the nomenclature I've adopted is a spatial array of neurons with identical stimulus selectivity, whereas each of the input neurons in this network responded preferentially to different depths at the same spatial location.) Both sets of input neurons had the same tuning curves, but responded to each of two different cues (e.g., stereo and motion parallax). The output neurons had the same tuning curves as the input neurons because the feedforward drive depended on a weighted sum of the responses of input neurons with identical tuning curves, with weights w_1 and w_2 (Eq. **S24**, 3rd line).

Each tuning curve, denoted $\psi_n(s)$, where s is stimulus depth, was one cycle of a raised cosine, and the spacing, amplitudes, and widths of the raised cosines were chosen so that the tuning curves summed to 1:

$$\psi_n(s)\propto\cos\left[\frac{2\pi\left(s-s_n\right)}{\nu}\right]+1$$
$$\text{for }-\pi<\frac{2\pi\left(s-s_n\right)}{\nu}<\pi\qquad\textbf{[S25]}$$
$$\sum_j\psi_n(s)=1,$$

The value of ν determined the width of the tuning curves, and the values of s_n determined the preferred depths (the peaks of the tuning curves). The preferred depths were evenly spaced and the widths were selected to be even multiples of the spacing. The spacing and width also determined the amount of overlap, overlap = spacing / 2ν; the overlap was 4 for the simulation results in Fig. 5 of the main text.

The responses of the input neurons depended on the strengths of the two cues (g_1 and g_2), and the responses of the input neurons were presumed to be noisy (additive, independent, normally-distributed):

$$y_{n1}^{(0)}\sim N\left(g_1\psi_n(s),\sigma_1^2\right)$$
$$y_{n2}^{(0)}\sim N\left(g_2\psi_n(s),\sigma_2^2\right),\qquad\textbf{[S26]}$$

where σ_1 and σ_2 are the standard deviations of the noise. If $\sigma_1=\sigma_2$ then the two cues were equally reliable; otherwise not.

The responses of the output neurons were modeled as dynamical processes (Eq. **S6**) that minimized this energy function (Eq. **S24**) over time, subject to $y_n^{(1)}\geq 0$.

The prior for the response of the n^{th} output neuron was defined in terms of the tuning curves. The two example priors shown in Fig. 5C of the main text corresponded to $\hat{y}_n^{(1)}=\psi_n(0)$ and $\hat{y}_n^{(1)}=\psi_n(-0.5)+\psi_n(0.5)$. Each of these priors for the responses of the output neurons conferred a prior over stimuli:

$$p_0(s)\propto\exp\left[-\frac{1}{2\sigma_0^2}\sum_n\left(\psi_n(s)-\hat{y}_n^{(1)}\right)^2\right],\qquad\textbf{[S27]}$$

where σ_0 specified the reliability of the prior.

The readout was defined as:

$$h\left(s\mid\mathbf{y}^{(1)}\right)=\exp\left[-\sum_n h_n\left(s\mid y_n^{(1)}\right)\right]$$
$$h_n\left(s\mid y_n^{(1)}\right)=\frac{1}{2}\alpha\lambda\left(y_n^{(1)}-\hat{g}\psi_n(s)\right)^2\qquad\textbf{[S28]}$$
$$+\frac{1}{2}\alpha(1-\lambda)\left(\frac{y_n^{(1)}}{\hat{g}}-\psi_n(s)\right)^2.$$

The readout h transformed the vector of responses of the output neurons to a continuous function of s that was approximately proportional to the Bayes-optimal posterior (as derived below). A variant of the readout computed a depth estimate and an uncertainty:

$$\hat{s}=\frac{\sum_k s_k h\left(s_k\mid\mathbf{y}^{(1)}\right)}{\sum_k h\left(s_k\mid\mathbf{y}^{(1)}\right)}\qquad\textbf{[S29]}$$
$$\sigma_{\hat{s}}^2=\frac{\sum_k\left(s_j-\hat{s}\right)^2 h\left(s_k\mid\mathbf{y}^{(1)}\right)}{\sum_k h\left(s_k\mid\mathbf{y}^{(1)}\right)},$$

where the depth estimate (or percept) $\hat{s}$ was approximately equal to the mean of the posterior, and the uncertainty $\sigma_{\hat{s}}$ was approximately equal to the standard deviation of the posterior. The value of k indexes a finite number of samples of s. Both variants of the readout (Eqs. **S28** and **S29**) depended only the responses of the output neurons $y_n^{(1)}$, the tuning curves $\psi_n(s)$, and the values of the state parameters λ and α. The Bayes-optimal posterior, on the other hand, depends on the responses of the input neurons $y_{n1}^{(0)}$ and $y_{n2}^{(0)}$, the noise standard deviations σ_1 and σ_2, the prior over stimuli $p_0(s)$, and the reliability of the prior σ_0.

Next I show that the readout is approximately proportional to the Bayes-optimal posterior, if the values of the state parameters are chosen correctly. There are two limiting cases corresponding to: 1) when the stimulus strengths of both cues are small, and 2) when the stimulus strengths of one or both cues are large. To begin, we need expressions for the probability distribution of z_n, and for the values of the state parameters.

The values of z_n were normally distributed because they were computed as weighted sums of normally-distributed random variables (Eq. **S24**, 3rd line):

$$z_n^{(1)} \sim N\left(g\psi_n(s), \sigma^2\right) \qquad \textbf{[S30]}$$

$$g = w_1 g_1 + w_2 g_2 \qquad \text{(see Eq. } \textbf{S24}\text{)}$$

$$\sigma^2 = w_1^2\sigma_1^2 + w_2^2\sigma_2^2 = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2} . \qquad \text{(see Eq. } \textbf{S24}\text{)}$$

The state parameters were chosen based on the reliabilities of each of the two cues and the reliability of the prior:

$$\alpha = r_0 + r_1 + r_2 \qquad \textbf{[S31]}$$

$$\lambda = \frac{r_1 + r_2}{r_0 + r_1 + r_2}$$

$$r_0 = \frac{1}{\sigma_0^2} \quad r_1 = \frac{1}{\sigma_1^2} \quad r_2 = \frac{1}{\sigma_2^2} .$$

where r_1 and r_2 are the reliabilities of each of the two cues, and r_0 is the reliability of the prior. For these values of the state parameters:

$$\alpha\lambda = \frac{1}{\sigma^2} \qquad \textbf{[S32]}$$

$$\alpha(1-\lambda) = \frac{1}{\sigma_0^2} .$$

It is helpful to rewrite the readout:

$$h\left(s \mid \mathbf{y^{(1)}}\right) = \exp\left[-\frac{1}{2}\alpha\lambda\sum_n\left(y_n^{(1)} - \hat{g}\psi_n(s)\right)^2 - \frac{1}{2}\alpha(1-\lambda)\sum_n\left(\frac{y_n^{(1)}}{\hat{g}} - \psi_n(s)\right)^2\right]$$

$$= \exp\left[-\frac{1}{2}\alpha\left(\lambda\hat{g}^2 + 1 - \lambda\right)\sum_j\left(\frac{y_n^{(1)}}{\hat{g}} - \psi_n(s)\right)^2\right] \qquad \textbf{[S33]}$$

Case 1: When the stimulus strengths of both cues are small, the readout is approximately proportional to the prior over s. If both cues are weak:

$$\hat{g}^2 \ll (1-\lambda) \qquad \textbf{[S34]}$$

$$\left(\lambda\hat{g}^2 + 1 - \lambda\right) = (1-\lambda)\left(\lambda\frac{\hat{g}^2}{1-\lambda} + 1\right) \approx (1-\lambda) .$$

In addition, when both stimulus strengths are weak, then the second term of E (Eq. **S24**) dominates and the responses converge to values that are proportional to the priors, i.e.,

$$\frac{y_n^{(1)}}{\hat{g}} \approx \hat{y}_n^{(1)} . \qquad \textbf{[S35]}$$

Consequently, the readout (Eq. **S33**) can be approximated:

$$\textbf{[S36]}$$

$$h\left(s \mid \mathbf{y^{(1)}}\right) \approx \exp\left[-\frac{1}{2}\alpha(1-\lambda)\sum_n\left(\frac{y_n^{(1)}}{\hat{g}} - \psi_n(s)\right)^2\right]$$

(see Eqs. **S33** and **S34**)

$$\approx \exp\left[-\frac{1}{2\sigma_0^2}\sum_n\left(\hat{y}_n^{(1)} - \psi_n(s)\right)^2\right]$$

(see Eqs. **S32** and **S35**)

$$\propto p_0(s) , \qquad \text{(see Eq. } \textbf{S27}\text{)}$$

where $p_0(s)$ is the prior over s.

Case 2: When the stimulus strengths of one or both cues are large, the readout is approximately proportional to the likelihood. If one or both cues are strong:

$$\hat{g}^2 \gg (1-\lambda) \qquad \textbf{[S37]}$$

$$\left(\lambda\hat{g}^2 + 1 - \lambda\right) = \hat{g}^2\left(\lambda + \frac{1-\lambda}{\hat{g}^2}\right) \approx \left(\lambda\hat{g}^2\right) .$$

In addition, when one or both cues are strong, then the first term of E (Eq. **S24**) dominates and the responses converge to minimize the feedforward drive, i.e.,

$$y_n^{(1)} \approx z_n^{(1)} , \qquad \textbf{[S38]}$$

And when one or both cues are strong, then $\hat{g} \approx g$:

$$\hat{g} = \sum_n y_n^{(1)} \approx \sum_n z_n^{(1)} \qquad \textbf{[S39]}$$

$$= \left(w_1\sum_n y_{n1}^{(0)} + w_2\sum_n y_{n2}^{(0)}\right) \qquad \text{(see Eq. } \textbf{S24}\text{)}$$

$$\approx \left(w_1 g_1\sum_n \psi_n(s) + w_2 g_2\sum_n \psi_n(s)\right)$$

(see Eq. **S26**)

$$= \left(w_1 g_1 + w_2 g_2\right) = g$$

(see Eqs. **S25** and **S30**)

Consequently, the readout (Eq. **S33**) can be approximated:

[S40]

$$h\left(s \mid \mathbf{y}^{(1)}\right) \approx \exp\left[-\frac{1}{2}\alpha\left(\lambda \hat{g}^2\right)\sum_j \left(\frac{y_j^{(1)}}{\hat{g}} - \psi_j(s)\right)^2\right]$$

(see Eqs. **S33** and **S37**)

$$= \exp\left[-\frac{1}{2}\alpha\lambda \sum_j \left(y_j^{(1)} - \hat{g}\psi_j(s)\right)^2\right]$$

$$\approx \exp\left[-\frac{1}{2\sigma^2}\sum_j \left(z_j^{(1)} - g\psi_j(s)\right)^2\right]$$

(see Eqs. **S32**, **S38**, **S39**)

$$\propto p\left(\mathbf{z}^{(1)} \mid s\right)$$

(see Eq. **S30**)

Finally, when one or both cues are strong, then $p(\mathbf{z}^{(1)} \mid s)$ is approximately proportional to the likelihood $p(\mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)} \mid s)$. The negative log likelihoods are:

[S41]

$$-\log\left[p\left(\mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)} \mid s\right)\right] = \sum_n \left[\frac{\left(y_{n1}^{(0)} - g_1\psi_n(s)\right)^2}{2\sigma_1^2} + \frac{\left(y_{n2}^{(0)} - g_2\psi_n(s)\right)^2}{2\sigma_2^2}\right] - \log(c_0)$$

$$-\log\left[p\left(\mathbf{z}^{(1)} \mid s\right)\right] = \sum_n \frac{\left(z_n^{(1)} - g\psi_n(s)\right)^2}{2\sigma^2} - \log(c_1),$$

where c_0 and c_1 are proportionality constants. It suffices to show that each of the terms in the summations are approximately equal to one another:

[S42]

$$\frac{\left(z_n^{(1)} - g\psi_n(s)\right)^2}{2\sigma^2} = \frac{\sigma_1^2 + \sigma_2^2}{2\sigma_1^2\sigma_2^2}\left(w_1 y_{n1}^{(0)} + w_2 y_{n2}^{(0)} - w_1 g_1 \psi_n(s) - w_2 g_2 \psi_n(s)\right)^2$$

$$= \frac{\left(y_{n1}^{(0)} - g_1\psi_n(s)\right)^2}{2\sigma_1^2} + \frac{\left(y_{n2}^{(0)} - g_2\psi_n(s)\right)^2}{2\sigma_2^2} + \frac{\left(y_{n1}^{(0)} - g_1\psi_n(s)\right)\left(y_{n2}^{(0)} - g_2\psi_n(s)\right)}{\sigma_1\sigma_2}$$

$$\approx \frac{\left(y_{n1}^{(0)} - g_1\psi_n(s)\right)^2}{2\sigma_1^2} + \frac{\left(y_{n2}^{(0)} - g_2\psi_n(s)\right)^2}{2\sigma_2^2}.$$

The last step relies on an approximation that the cross-term can be ignored. This approximation is reasonable when the stimulus strengths of one or both cues are large, specifically when either: $g_1/\sigma_1 \gg g_2/\sigma_2$, or $g_2/\sigma_2 \gg g_1/\sigma_1$, or both $g_1/\sigma_1 \gg 1$, and $g_2/\sigma_2 \gg 1$.

1. Lecun Y, *et al.* (1989) Backpropagation applied to handwritten zip code recognition. *Neural Comput* 1:541-551.
2. Ackley DH, Hinton GE, & Sejnowski TJ (1985) A learning algorithm for Boltzmann Machines. *Cognitive Science* 9(1):147-169.
3. Lewicki MS & Olshausen BA (1999) Probabilistic framework for the adaptation and comparison of image codes. *J Opt Soc Am A* 16(7):1587-1601.
4. Cadieu CF & Olshausen BA (2012) Learning intermediate-level representations of form and motion from natural movies. *Neural Comput* 24(4):827-866.
5. Mumford D (1992) On the computational architecture of the neocortex. II. The role of cortico-cortical loops. *Biol Cybern* 66(3):241-251.
6. Rumelhart DE, McClelland JL, & Group PR (1988) *Parallel distributed processing* (IEEE).
7. Hinton GE (2007) Learning multiple layers of representation. *Trends Cogn Sci* 11(10):428-434.
8. Freeman J & Simoncelli EP (2011) Metamers of the ventral stream. *Nat Neurosci* 14(9):1195-1201.
9. Friston K (2005) A theory of cortical responses. *Philos Trans R Soc Lond B Biol Sci* 360(1456):815-836.
10. Bastos AM, *et al.* (2012) Canonical microcircuits for predictive coding. *Neuron* 76(4):695-711.
11. Pineda FJ (1987) Generalization of back-propagation to recurrent neural networks. *Phys Rev Lett* 59(19):2229-2232.
12. Carpenter GA & Grossberg S (1987) A Massively Parallel Architecture for a Self-Organizing Neural Pattern-Recognition Machine. *Comput Vision Graph* 37(1):54-115.
13. Bruna J & Mallat S (2013) Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence* 35(8):1872-1886.
14. Girshick AR, Landy MS, & Simoncelli EP (2011) Cardinal rules: visual orientation perception reflects knowledge of environmental statistics. *Nat Neurosci* 14(7):926-932.
15. Knill DC & Richards W (1996) *Perception as Bayesian inference* (Cambridge University Press).
16. Weiss Y, Simoncelli EP, & Adelson EH (2002) Motion illusions as optimal percepts. *Nat Neurosci* 5(6):598-604.
17. Mamassian P, Landy MS, & Maloney LT (2002) Bayesian modelling of visual perception. *Probabilistic models of the brain: Perception and neural function*:13-36.
18. Yuille A & Kersten D (2006) Vision as Bayesian inference: analysis by synthesis? *Trends Cogn Sci* 10(7):301-308.
19. Brainard DH & Freeman WT (1997) Bayesian color constancy. *J Opt Soc Am A* 14(7):1393-1411.
20. Laing CR & Chow CC (2002) A spiking neuron model for binocular rivalry. *Journal of computational neuroscience* 12(1):39-53.
21. Moreno-Bote R, Rinzel J, & Rubin N (2007) Noise-induced alternations in an attractor network model of perceptual bistability. *J Neurophysiol* 98(3):1125-1139.
22. Wilson HR (2003) Computational evidence for a rivalry hierarchy in vision. *Proc Natl Acad Sci U S A* 100(24):14499-14503.
23. Wilson HR (2007) Minimal physiological conditions for binocular rivalry and rivalry memory. *Vision Res* 47(21):2741-2750.
24. Jordan MI & Rumelhart DE (1992) Forward models: Supervised learning with a distal teacher. *Cognitive Science* 16(3):307-354.
25. Riesenhuber M & Poggio T (2002) Neural mechanisms of object recognition. *Curr Opin Neurobiol* 12(2):162-168.
26. Fleet DJ, Wagner H, & Heeger DJ (1996) Neural encoding of binocular disparity: energy models, position shifts and phase shifts. *Vision Res* 36(12):1839-1857.
27. Landy MS & Bergen JR (1991) Texture segregation and orientation gradient. *Vision Res* 31(4):679-691.
28. Zipser D, Kehoe B, Littlewort G, & Fuster J (1993) A spiking network model of short-term active memory. *J Neurosci* 13(8):3406-3420.
29. Seung HS (1996) How the brain keeps the eyes still. *Proc Natl Acad Sci U S A* 93(23):13339-13344.
30. Zhang K (1996) Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory. *J Neurosci* 16(6):2112-2126.

31. Robinson DA (1989) Integrating with neurons. *Annual review of neuroscience* 12:33-45.

32. Major G & Tank D (2004) Persistent neural activity: prevalence and mechanisms. *Curr Opin Neurobiol* 14(6):675-684.

33. Li Z (1998) A neural model of contour integration in the primary visual cortex. *Neural Comput* 10(4):903-940.

34. Li Z (1999) Visual segmentation by contextual influences via intracortical interactions in the primary visual cortex. *Network* 10(2):187-212.

35. Li Z (1999) Contextual influences in V1 as a basis for pop out and asymmetry in visual search. *Proc Natl Acad Sci U S A* 96(18):10530-10535.

36. Coen-Cagli R, Dayan P, & Schwartz O (2012) Cortical Surround Interactions and Perceptual Salience via Natural Scene Statistics. *PLoS Comput Biol* 8(3):e1002405.

37. Freeman WT & Adelson EH (1991) The design and use of steerable filters. *IEEE Pattern Analysis and Machine Intelligence* 13:891-906.

38. Simoncelli EP, Freeman WT, Adelson EH, & Heeger DJ (1992) Shiftable multi-scale transforms. *IEEE Transactions on Information Theory, Special Issue on Wavelets* 38:587-607.

39. Simoncelli EP (1993) Distributed representation and analysis of visual motion. (Massachusetts Institute of Technology, Cambridge, MA).

40. Carandini M & Heeger DJ (2012) Normalization as a canonical neural computation. *Nature reviews. Neuroscience* 13(1):51-62.

41. Heeger DJ (1992) Half-squaring in responses of cat striate cells. *Vis Neurosci* 9(5):427-443.

42. Heeger DJ (1992) Normalization of cell responses in cat striate cortex. *Vis Neurosci* 9(2):181-197.

43. Zoccolan D, Cox DD, & DiCarlo JJ (2005) Multiple object response normalization in monkey inferotemporal cortex. *J Neurosci* 25(36):8150-8164.

44. Rabinowitz NC, Willmore BD, Schnupp JW, & King AJ (2011) Contrast gain control in auditory cortex. *Neuron* 70(6):1178-1191.

45. Ohshiro T, Angelaki DE, & Deangelis GC (2011) A normalization model of multisensory integration. *Nature neuroscience* 14(6):775-782.

46. Louie K & Glimcher PW (2010) Separating value from choice: delay discounting activity in the lateral intraparietal area. *J Neurosci* 30(16):5498-5507.

47. Vokoun CR, Huang X, Jackson MB, & Basso MA (2014) Response normalization in the superficial layers of the superior colliculus as a possible mechanism for saccadic averaging. *J Neurosci* 34(23):7976-7987.

48. Olsen SR, Bhandawat V, & Wilson RI (2010) Divisive normalization in olfactory population codes. *Neuron* 66(2):287-299.

49. Herrmann K, Montaser-Kouhsari L, Carrasco M, & Heeger DJ (2010) When size matters: attention affects performance by contrast or response gain. *Nat Neurosci* 13(12):1554-1559.

50. Itthipuripat S, Garcia JO, Rungratsameetaweemana N, Sprague TC, & Serences JT (2014) Changing the spatial scope of attention alters patterns of neural gain in human cortex. *J Neurosci* 34(1):112-123.

51. Reynolds JH & Heeger DJ (2009) The normalization model of attention. *Neuron* 61(2):168-185.

52. Sundberg KA, Mitchell JF, & Reynolds JH (2009) Spatial attention modulates center-surround interactions in macaque visual area v4. *Neuron* 61(6):952-963.

53. Brouwer GJ & Heeger DJ (2011) Cross-orientation suppression in human visual cortex. *Journal of neurophysiology* 106(5):2108-2119.

54. Brouwer GJ, Arnedo V, Offen S, Heeger DJ, & Grant AC (2015) Normalization in human somatosensory cortex. *J Neurophysiol* 114(5):2588-2599.

55. Carandini M, Heeger DJ, & Movshon JA (1997) Linearity and normalization in simple cells of the macaque primary visual cortex. *J Neurosci* 17(21):8621-8644.

56. Cavanaugh JR, Bair W, & Movshon JA (2002) Nature and interaction of signals from the receptive field center and surround in macaque V1 neurons. *J Neurophysiol* 88(5):2530-2546.

57. Cavanaugh JR, Bair W, & Movshon JA (2002) Selectivity and spatial distribution of signals from the receptive field surround in macaque V1 neurons. *J Neurophysiol* 88(5):2547-2556.

58. Smith MA, Bair W, & Movshon JA (2006) Dynamics of suppression in macaque primary visual cortex. *J Neurosci* 26(18):4826-4834.

59. Zenger-Landolt B & Heeger DJ (2003) Response suppression in V1 agrees with psychophysics of surround masking. *J Neurosci* 23(17):6884-6893.

60. Simoncelli EP & Heeger DJ (1998) A model of neuronal responses in visual area MT. *Vision Res* 38(5):743-761.

61. Foley JM (1994) Human luminance pattern-vision mechanisms: masking experiments require a new model. *J Opt Soc Am A* 11(6):1710-1719.

62. Xing J & Heeger DJ (2000) Center-surround interactions in foveal and peripheral vision. *Vision Res* 40(22):3065-3072.

63. Xing J & Heeger DJ (2001) Measurement and modeling of center-surround suppression and enhancement. *Vision Res* 41(5):571-583.

64. Carandini M & Ferster D (2000) Membrane potential and firing rate in cat primary visual cortex. *J Neurosci* 20(1):470-484.

65. Carandini M (2007) Melting the iceberg: contrast invariance in visual cortex. *Neuron* 54(1):11-13.

66. Carandini M & Heeger DJ (1994) Summation and division by neurons in primate visual cortex. *Science* 264(5163):1333-1336.

67. Heeger DJ (1993) Modeling simple-cell direction selectivity with normalized, half- squared, linear operators. *J Neurophysiol* 70(5):1885-1898.

68. Rubin DB, Van Hooser SD, & Miller KD (2015) The stabilized supralinear network: a unifying circuit motif underlying multi-input integration in sensory cortex. *Neuron* 85(2):402-417.

69. Olsen SR & Wilson RI (2008) Lateral presynaptic inhibition mediates gain control in an olfactory circuit. *Nature* 452(7190):956-960.

70. Katzner S, Busse L, & Carandini M (2011) GABAA inhibition controls response gain in visual cortex. *J Neurosci* 31(16):5931-5941.

71. Sato TK, Haider B, Hausser M, & Carandini M (2016) An excitatory basis for divisive normalization in visual cortex. *Nat Neurosci*.

72. Nowlan SJ & Sejnowski TJ (1995) A selection model for motion processing in are MT of primates. *Journal of Neuroscience* 15:1195-1214.

73. Kouh M & Poggio T (2008) A canonical neural circuit for cortical nonlinear operations. *Neural Comput* 20(6):1427-1451.

74. Ganguli D & Simoncelli EP (2014) Efficient sensory encoding and Bayesian inference with heterogeneous neural populations. *Neural Comput* 26(10):2103-2134.

75. Sarter M & Bruno JP (1997) Cognitive functions of cortical acetylcholine: toward a unifying hypothesis. *Brain Res Brain Res Rev* 23(1-2):28-46.

76. Yu AJ & Dayan P (2005) Uncertainty, neuromodulation, and attention. *Neuron* 46(4):681-692.

77. Marshall L*, et al.* (2016) Pharmacological Fingerprints of Contextual Uncertainty. *PLoS Biol* 14(11):e1002575.

78. Pafundo DE, Nicholas MA, Zhang R, & Kuhlman SJ (2016) Top-Down-Mediated Facilitation in the Visual Cortex Is Gated by Subcortical Neuromodulation. *J Neurosci* 36(10):2904-2914.

79. Schultz W, Dayan P, & Montague PR (1997) A neural substrate of prediction and reward. *Science* 275(5306):1593-1599.

80. McGinley MJ, David SV, & McCormick DA (2015) Cortical Membrane Potential Signature of Optimal States for Sensory Signal Detection. *Neuron* 87(1):179-192.

81. Pisauro MA, Dhruv NT, Carandini M, & Benucci A (2013) Fast hemodynamic responses in the visual cortex of the awake mouse. *J Neurosci* 33(46):18343-18351.

82. Einhauser W, Stout J, Koch C, & Carter O (2008) Pupil dilation reflects perceptual selection and predicts subsequent stability in perceptual rivalry. *Proc Natl Acad Sci U S A* 105(5):1704-1709.

83. Aston-Jones G & Cohen JD (2005) An integrative theory of locus coeruleus-norepinephrine function: adaptive gain and optimal performance. *Annual review of neuroscience* 28:403-450.

84. Teigen KH (1994) Yerkes-Dodson: A law for all seasons. *Theory & Psychology* 4(4):525-547.

85. Yerkes RM & Dodson JD (1908) The Relation of Strength of Stimulus to Rapidity of Habit Formation. *Journal of Comparative Neurology & Psychology*:459-482.

86. Gilden DL (2001) Cognitive emissions of 1/f noise. *Psychol Rev* 108(1):33-56.

87. Gilden DL, Thornton T, & Mallon MW (1995) 1/f noise in human cognition. *Science* 267(5205):1837-1839.

88. Rigoli LM, Holman D, Spivey MJ, & Kello CT (2014) Spectral convergence in tapping and physiological fluctuations: coupling and independence of 1/f noise in the central and autonomic nervous systems. *Front Hum Neurosci* 8:713.

89. VanRullen R (2016) Perceptual Cycles. *Trends Cogn Sci* 20(10):723-735.

90. Adelson EH & Bergen JR (1985) Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America A* 2:284-299.